UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/649,307 | 08/26/2003 | Masanori Honda | 16869N-091900US | 5407 |

20350 7590 01/31/2006

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

| EXAMINER | |
|---|---|
| TAN, ALVIN H | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2173 | |

DATE MAILED: 01/31/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/649,307 | HONDA ET AL. | |
| | Examiner | Art Unit | |
| | Alvin H. Tan | 2173 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *20 December 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-6,8-14 and 16-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6, 8-14, 16-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

### *Response to Amendment*

1. Claims 1-6, 8-14, and 16-22 have been examined and rejected. This Office action is responsive to the amendment filed on 12/20/05, which has been entered in the above identified application.

### *Claim Rejections - 35 USC § 112*

2. The corrections to claims 1-14 and 16-20 have been approved, and the rejections to the claims under 35 U.S.C § 112, first paragraph, are withdrawn.

3. The corrections to claims 12 and 14 have been approved, and the rejections to the claims under 35 U.S.C § 112, second paragraph, are withdrawn.

### *Claim Rejections - 35 USC § 103*

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-5, 8-14, and 16-22 are rejected under 35 U.S.C. 102(b) as being anticipated by Park et al (Pub. No US 2001/0039594 A1).

**Claims 1-5, 8, 9**

5-1. Regarding claim 1, Park teaches the claim of a job management method comprising storing a stencil for a job definition statement, by disclosing a template script contained within a workflow template file *[paragraph 56, lines 5-7]* along with workflow markup that sets the workflow rules *[paragraph 44, lines 1-3]* used to create a job specification file *[paragraph 41, lines 4-7]*.

Park teaches data prescribing a user interface for job definition statement setup, by disclosing that a job may be created through end-user input from a browser interface *[paragraph 41, lines 4-7]*.

Park teaches wherein the data prescribing a user interface for job definition statement setup is used to produce a user interface for generating a job definition statement, by disclosing that the browser creates one or more workflow forms *[paragraph 52, lines 1-4]*, which are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification *[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches generating data for executing a process for generating a job definition statement based on contents set by a user via the user interface in accordance with the stencil for the job definition statement and the data prescribing the user interface for job definition statement setup and generating the job definition

statement by executing the process in accordance with the generated data, by disclosing a technique for creating a job that automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 15-18]* by using a workflow template file (containing a template script and workflow markup) along with user input *[paragraph 41, lines 4-7]*.

Park teaches that the stencil and the data prescribing the user interface for job definition statement setup are stored in the computer-readable storage medium, by disclosing that a development server includes a conventional memory and a conventional processor which implements the website development methods of the present invention by executing software stored in memory *[paragraph 21, lines 1-5]*. Thus, the workflow template files, scripts, and forms are stored in memory, prior for generating data for executing the process.

Park teaches that a template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil.

However, examiner takes Official Notice that the Perl programming language contains conditional statements, which can be used to invalidate a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the conditional statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains conditional statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations with conditional statements to control the workflow based on user input would be beneficial.

5-2. Regarding claim 2, Park teaches the claim of the method wherein the stencil for the job definition statement is written in XML format, by disclosing that the workflow template file containing the workflow markup and template script elements used to define the workflow markups is an XML file *[paragraph 41, lines 2-3]*.

5-3. Regarding claim 3, Park teaches the claim of the method wherein the data prescribing the user interface for setting the job definition statement is written in XML format and positioned in a file in which the stencil for the job definition statement is written, by disclosing that a job may be created through a combination of workflow rules defined in a workflow template file (e.g., an XML file) and end-user input from the browser interface. The workflow rules of the template and user input may be interpreted by a common gateway interface to dynamically create a job specification file *[paragraph 41, lines 4-13]*. Based on the workflow markup, the instantiator CGI creates one or more workflow forms into which a user can enter workflow configuration information via the browser *[paragraph 52, lines 1-4]*. The template script and workflow markup that make up the stencil are contained within the workflow template file *[paragraph 56, lines 5-7]*.

5-4. Regarding claim 4, Park teaches the claim of the method wherein the user interface is capable of opening a window, which is organized to prompt a user for setup, in order to prompt the user to set the job definition statement, by teaching a browser interface GUI that lets a user select a workflow template, such as from a menu item *[paragraph 50, lines 1-3]*.

5-5. Regarding claim 5, Park teaches the claim of the method wherein the data prescribing the user interface for setting the job definition statement contains control data for specifying whether the window should display user-definable options, by disclosing that a workflow template file can contain <template_script> elements,

including a set of directives to define the workflow markups *[paragraph 56, lines 5-7]*. The template scripts may modify the workflow forms, onto which a user can enter configuration information to create a job specification *[paragraph 52, lines 1-4]*.

5-6. Regarding claim 8, Park teaches that the template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement issuing an instruction for generating a job definition statement in which a specific description written in the stencil is repeatedly written.

However, examiner takes Official Notice that the Perl programming language contains loop statements, which can be used to repeatedly write a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the loop statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language

contains loop statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. Thus, a loop statement would be a more efficient way of repeatedly writing a specific description in the stencil.

5-7. Regarding claim 9, Park teaches the claim of the method wherein the job definition statement contains a definition statement for setting a job network that executes a stream of a plurality of jobs, by disclosing a <submittask> element that may perform a submit operation (to a staging area) on its contained files. If successful, specified successor tasks may be signaled *[paragraph 37, lines 14-16]*. Thus, a plurality of jobs would be executed after the current task is completed.

**Claims 10-11, 21**

5-8. Regarding claim 10, it is similar to claim 1 except that it claims a system instead of a method. Therefore, it is rejected under the same reasons as claim 1 above. See section 5-1.

Park teaches the claim of a system comprising storing a stencil for a job definition statement, by disclosing a template script contained within a workflow template file *[paragraph 56, lines 5-7]* along with workflow markup that sets the workflow rules *[paragraph 44, lines 1-3]* used to create a job specification file *[paragraph 41, lines 4-7]*.

Park teaches data prescribing a user interface for job definition statement setup, by disclosing that a job may be created through end-user input from a browser interface *[paragraph 41, lines 4-7]*.

Park teaches wherein the data prescribing a user interface for job definition statement setup is used to produce a user interface for generating a job definition statement, by disclosing that the browser creates one or more workflow forms *[paragraph 52, lines 1-4]*, which are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification *[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches generating data for executing a process for generating a job definition statement based on contents set by a user via the user interface in accordance with the stencil for the job definition statement and the data prescribing the user interface for job definition statement setup and generating the job definition statement by executing the process in accordance with the generated data, by disclosing a technique for creating a job that automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 15-18]* by using a workflow template file (containing a template script and workflow markup) along with user input *[paragraph 41, lines 4-7]*.

Park teaches that the stencil and the data prescribing the user interface for job definition statement setup are stored in the computer-readable storage medium, by

disclosing that a development server includes a conventional memory and a conventional processor which implements the website development methods of the present invention by executing software stored in memory *[paragraph 21, lines 1-5]*. Thus, the workflow template files, scripts, and forms are stored in memory, prior for generating data for executing the process.

Park teaches that a template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil.

However, examiner takes Official Notice that the Perl programming language contains conditional statements, which can be used to invalidate a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the conditional statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language

contains conditional statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations with conditional statements to control the workflow based on user input would be beneficial.

5-9. Regarding claim 11, Park teaches the claim of the system further comprising means for importing or exporting the stencil for the job definition statement that is managed as data in file form, by disclosing that a job may be created through a combination of workflow rules defined in a workflow template file (e.g., an XML file) and end-user input from the browser interface. The workflow rules of the template and user input may be interpreted by a common gateway interface to dynamically create a job specification file. Once configured, this technique simplifies the process of defining jobs since it provides a browser interface for user input and automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 4-18]*.

5-10. Regarding claim 21, Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user

interface, a definition statement issuing an instruction for generating a job definition statement in which a specific description written in the stencil is repeatedly written.

However, examiner takes Official Notice that the Perl programming language contains loop statements, which can be used to repeatedly write a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the loop statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains loop statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. Thus, a loop statement would be a more efficient way of repeatedly writing a specific description in the stencil.

**Claim 12, 22**

5-11. Regarding claim 12, it is similar to claim 1 except that it claims a computer readable medium comprising code instead of a method. Therefore, it is rejected under the same reasons as claim 1 above. See section 5-1.

Park teaches the claim of a computer readable medium having a program comprising code for storing a stencil for a job definition statement, by disclosing a template script contained within a workflow template file *[paragraph 56, lines 5-7]* along with workflow markup that sets the workflow rules *[paragraph 44, lines 1-3]* used to create a job specification file *[paragraph 41, lines 4-7]*.

Park teaches data prescribing a user interface for job definition statement setup, by disclosing that a job may be created through end-user input from a browser interface *[paragraph 41, lines 4-7]*.

Park teaches wherein the data prescribing a user interface for job definition statement setup is used to produce a user interface for generating a job definition statement, by disclosing that the browser creates one or more workflow forms *[paragraph 52, lines 1-4]*, which are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification *[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches generating data for executing a process for generating a job definition statement based on contents set by a user via the user interface in accordance with the stencil for the job definition statement and the data prescribing the

user interface for job definition statement setup and generating the job definition statement by executing the process in accordance with the generated data, by disclosing a technique for creating a job that automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 15-18]* by using a workflow template file (containing a template script and workflow markup) along with user input *[paragraph 41, lines 4-7]*.

Park teaches that the stencil and the data prescribing the user interface for job definition statement setup are stored in the computer-readable storage medium, by disclosing that a development server includes a conventional memory and a conventional processor which implements the website development methods of the present invention by executing software stored in memory *[paragraph 21, lines 1-5]*. Thus, the workflow template files, scripts, and forms are stored in memory, prior for generating data for executing the process.

Park teaches that a template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil.

However, examiner takes Official Notice that the Perl programming language contains conditional statements, which can be used to invalidate a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the conditional statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains conditional statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations with conditional statements to control the workflow based on user input would be beneficial.

5-12. Regarding claim 22, Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement issuing an instruction for generating a job definition statement in which a specific description written in the stencil is repeatedly written.

However, examiner takes Official Notice that the Perl programming language contains loop statements, which can be used to repeatedly write a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the loop statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains loop statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. Thus, a loop statement would be a more efficient way of repeatedly writing a specific description in the stencil.

**Claim 13**

5-13. Regarding claim 13, Park teaches a computer readable storage medium having a program comprising code for generating a job definition statement based on contents set by a user via a user interface in accordance with a stencil for the job definition statement and data prescribing the user interface for job definition statement setup, by

disclosing a technique for creating a job that automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 15-18]* by using a workflow template file (containing a template script) along with user input *[paragraph 41, lines 4-7]*.

Park teaches data prescribing the user interface for job definition statement setup, by disclosing that a job may be created through end-user input from a browser interface *[paragraph 41, lines 4-7]*.

Park teaches that the stencil and the data prescribing the user interface for job definition statement setup are stored in a computer-readable storage medium, by disclosing that a development server includes a conventional memory and a conventional processor which implements the website development methods of the present invention by executing software stored in memory *[paragraph 21, lines 1-5]*. Thus, the workflow template files, scripts, and forms are stored in memory, prior for generating data for executing the process.

Park teaches wherein the data prescribing a user interface for job definition statement setup is used to produce a user interface for generating a job definition statement, by disclosing that the browser creates one or more workflow forms *[paragraph 52, lines 1-4]*, which are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification *[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches that a template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil.

However, examiner takes Official Notice that the Perl programming language contains conditional statements, which can be used to invalidate a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the conditional statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains conditional statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations

with conditional statements to control the workflow based on user input would be beneficial.

**Claim 14**

5-14. Regarding claim 14, it is similar to claim 1 except that it claims a computer readable medium comprising means for claim 1 instead of a method. Therefore, it is rejected under the same reasons as claim 1 above. See section 5-1.

Park teaches the claim of a computer readable medium having a program for storing a stencil for a job definition statement, by disclosing a template script contained within a workflow template file *[paragraph 56, lines 5-7]* along with workflow markup that sets the workflow rules *[paragraph 44, lines 1-3]* used to create a job specification file *[paragraph 41, lines 4-7]*.

Park teaches data prescribing a user interface for job definition statement setup, by disclosing that a job may be created through end-user input from a browser interface *[paragraph 41, lines 4-7]*.

Park teaches wherein the data prescribing a user interface for job definition statement setup is used to produce a user interface for generating a job definition statement, by disclosing that the browser creates one or more workflow forms *[paragraph 52, lines 1-4]*, which are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification

*[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches generating data for executing a process for generating a job definition statement based on contents set by a user via the user interface in accordance with the stencil for the job definition statement and the data prescribing the user interface for job definition statement setup and generating the job definition statement by executing the process in accordance with the generated data, by disclosing a technique for creating a job that automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 15-18]* by using a workflow template file (containing a template script and workflow markup) along with user input *[paragraph 41, lines 4-7]*.

Park teaches that the stencil and the data prescribing the user interface for job definition statement setup are stored in a computer-readable storage medium, by disclosing that a development server includes a conventional memory and a conventional processor which implements the website development methods of the present invention by executing software stored in memory *[paragraph 21, lines 1-5]*. Thus, the workflow template files, scripts, and forms are stored in memory, prior for generating data for executing the process.

Park teaches that a template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil.

However, examiner takes Official Notice that the Perl programming language contains conditional statements, which can be used to invalidate a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the conditional statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains conditional statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations with conditional statements to control the workflow based on user input would be beneficial.

**Claims 16-20**

5-15. Regarding claim 16, Park teaches a storage system which includes a storage device *[figure 1, reference character 112]* for storing data for use in an operation server and a management server *[figure 1, reference character 104]* for managing the operation of the storage device, defining a job targeted for execution as a policy rule, and executing a process in compliance with the policy rule (server "104" implements the methods of the invention disclosed by Park *[paragraph 21, lines 1-4]*), the storage system comprising a storage section configured to store information about data concerning the policy rule and data prescribing a user interface for setting the policy rule, by disclosing that a job specification file may describe a single job and may be stored in the memory 112 *[paragraph 34, lines 5-7]*.

Park teaches a policy wizard GUI (browser interface used to create a job specification file *[paragraph 50, lines 1-3]*) which is configured to read the element attribute information about a policy rule from the storage section, process an element of a wizard page defining a guidance window for policy setup, and generate a wizard window, by disclosing that based upon the workflow markup, the instantiator CGI creates one or more workflow forms into which a user can enter workflow configuration information via the browser *[paragraph 52, lines 1-4]*. The workflow forms are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification *[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches the policy wizard GUI configured to enter policy rule setup information via the user interface in compliance with an instruction displayed by the generated window, by disclosing that the user enters workflow configuration information via the browser *[paragraph 52, lines 1-4]*, which is verified and if the input does not meet all necessary criteria, the interface re-prompts the user so that data can be re-entered *[paragraph 53, lines 1-7]*, the configuration information used to create a job specification file *[paragraph 54, lines 1-5]*.

Park teaches the policy wizard GUI configured to generate a policy rule in accordance with the information entered via the user interface, by disclosing that the browser interface is used to create a job specification file *[paragraph 54, lines 1-5]*.

Park teaches that a template script can insert text into a job specification programmatically *[paragraph 59, lines 6-7]* using the Perl programming language *[paragraph 58, line 2]*.

Park does not expressly disclose the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil.

However, examiner takes Official Notice that the Perl programming language contains conditional statements, which can be used to invalidate a specific description or portion of text.

Since the template script defines the workflow markups *[paragraph 56, lines 6-7]* used to create workflow forms into which a user enters workflow configuration information via a browser *[paragraph 52, lines 1-4]*, it would have been obvious to one

of ordinary skill in the art at the time the invention was made to include a definition statement for invalidating a specific description in the stencil by using the conditional statements provided by the Perl programming language in order to allow the user to have better control of text inserted into the job specification programmatically, since examiner takes Official Notice that it is well known that the Perl programming language contains conditional statements. The invention of Park relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations with conditional statements to control the workflow based on user input would be beneficial.

5-16. Regarding claim 17, Park teaches the claim of the system wherein the policy rule set via the user interface is stored in XML format in the storage section, by disclosing that the job specification file is in XML format *[paragraph 41, lines 2-3]*.

5-17. Regarding claim 18, Park teaches the claim of the system wherein a policy template, defining data for executing a policy rule generation process in accordance with contents set by a user via the user interface, contains information about a policy guidance window serving as the user interface and information about a generated policy definition XML file, by disclosing a workflow template file contains any or all of the

elements that are valid in a job specification file. These elements form the set of general workflow configuration instructions that are used when the user inputs configuration information via the browser *[paragraph 56, lines 1-5]*.

5-18. Regarding claim 19, Park teaches the claim of the system wherein the policy rule generated by the policy wizard GUI is delivered to a policy execution engine and registered, by disclosing that once a job specification is created, it is instantiated into the server and started in the server-side workflow subsystem *[paragraph 54, lines 8-10]*.

5-19. Regarding claim 20, Park teaches the claim of the system wherein the storage section stores beforehand information about instances of data concerning the policy rule and information about instances of data prescribing a user interface for setting the policy rule, by disclosing that a user may select a workflow template for defining a job *[paragraph 50, lines 1-3]*.

### *Claim Rejections - 35 USC § 103*

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Park et al (Pub No US 2001/0039594 A1) and Internet Explorer, as taught by McFedries (1997).

**Claim 6**

7-1. Regarding claim 6, Park teaches the invention substantially as claimed. See sections 5-1 and 5-4. Park teaches the claim of a job management method comprising storing a stencil for a job definition statement, by teaching a template script contained within a workflow template file *[paragraph 56, lines 5-7]* along with workflow markup that sets the workflow rules *[paragraph 44, lines 1-3]* used to create a job specification file *[paragraph 41, lines 4-7]*.

Park teaches data prescribing a user interface for job definition statement setup, by disclosing that a job may be created through end-user input from a browser interface *[paragraph 41, lines 4-7]*.

Park teaches wherein the data prescribing a user interface for job definition statement setup is used to produce a user interface for generating a job definition statement, by disclosing that the browser creates one or more workflow forms *[paragraph 52, lines 1-4]*, which are created based on information in a workflow template file *[paragraph 49, lines 2-5]*. Data is entered by users based on the rules in the workflow template file and configuration instructions to create a job specification *[paragraph 49, lines 5-11]*. Thus, the workflow template file contains data prescribing a user interface for job definition statement setup.

Park teaches generating data for executing a process for generating a job definition statement based on contents set by a user via the user interface in accordance with the stencil for the job definition statement and the data prescribing the user interface for job definition statement setup and generating the job definition statement by executing the process in accordance with the generated data, by disclosing a technique for creating a job that automatically generates separate job specification files for each distinct set of user inputs *[paragraph 41, lines 15-18]* by using a workflow template file (containing a template script and workflow markup) along with user input *[paragraph 41, lines 4-7]*.

Park teaches that the stencil and the data prescribing the user interface for job definition statement setup are stored in a computer-readable storage medium, by disclosing that a development server includes a conventional memory and a conventional processor which implements the website development methods of the present invention by executing software stored in memory *[paragraph 21, lines 1-5]*. Thus, the workflow template files, scripts, and forms are stored in memory, prior for generating data for executing the process.

Park teaches the claim of the method wherein the user interface is capable of opening a window, which is organized to prompt a user for setup, in order to prompt the user to set the job definition statement, by disclosing a browser interface GUI that lets a user select a workflow template, such as from a menu item *[paragraph 50, lines 1-3]*.

Park teaches that based on the workflow markup, the instantiator CGI creates one or more workflow forms into which a user can enter workflow configuration

information via the browser *[paragraph 52, lines 1-4]*. The workflow markup is written in XML *[paragraph 41, lines 5-6]*.

Park does not expressly disclose the data prescribing the user interface for setting the job definition statement contains control data for specifying whether or not to display a window that can open subsequently to a preceding window depending on a user response to the preceding window.

McFedries teaches that Internet Explorer is capable of opening a link in a new window *[page 356]*. Since Park teaches that the user interface includes using a browser window, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use Internet Explorer as the browser, since it is widely available to many users. Thus, the browser window would contain control data for specifying whether or not to display a window that can open subsequently to a preceding window depending on a user response to the preceding window.

### ***Response to Arguments***

8. The Examiner acknowledges the Applicants' amendments to claims 1, 10, 12, 13, 14, and 16, the cancellation of claim 7, and the addition of claims 21 and 22. Regarding independent claims 1, 10, 12, 13, 14, and 16, the Applicants allege that Park et al (Pub No US 2001/0039594) as described in the previous Office action, does not contain the proper motivation to teach wherein the stencil for the job definition statement contains, in accordance with a user selection made via the user interface, a definition statement for invalidating a specific description written in the stencil. The invention of

Park, however, relates to control and administration of workflow processes utilized for website development and maintenance. The workflow model is an arrangement of tasks to be performed *[paragraph 31]*. As disclosed in *[paragraph 58]*, Perl code may be used to inspect incoming data and execute coded rules based on this data. Thus, enforcing regulations with conditional statements to control the workflow based on user input would be beneficial.

Applicants state that dependent claims 2-6, 8, 9, 11, and 17-22, recite all the limitations of the independent claims, and thus, are allowable in view of the remarks set forth regarding independently amended claims 1, 10, 12, and 16. However, as discussed above, Park is considered to teach claims 1, 10, 12, and 16, and consequently, claims 2-6, 8, 9, 11, and 17-22 are rejected.

The Applicants' arguments have thus been considered, but are not persuasive.

***Conclusion***

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alvin H. Tan whose telephone number is 571-272-8595. The examiner can normally be reached between 8:30am-4:30pm, Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on 571-272-4048. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AHT
Assistant Examiner
Art Unit 2173